

Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Yrittäjyys ja sähköinen liiketoiminta

2015

Ilari Laivo

# PELIN KENTTÄSUUNNITTELU



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

Ilari Laivo

## PELIN KENTTÄSUUNNITTELU

Tämän opinnäytetyön tavoitteena on käyttää ilmaisohjelmia luonnollisen näköisen kentän mallintamiseen. Referenssikohteena on Turun asutusalueen, Jyrkkälän vieressä sijaitseva mäki nimeltä Pahanieniemi. Kohteen monimuotoinen luonto soveltui kokonsa puolesta hyvin tavoitteiden toteuttamiseen.

Kenttää luotiin kuvaamalla aluksi referenssikohdetta. Samalla kuvattiin myös kohteesta käytettyjen mallien tekstuureita. Seuraavaksi kuvat viimeisteltiin tekstuureiksi GIMP - kuvankäsittelyohjelmalla. Projektin aikana käytössä olivat diffuse-, normal- ja specular-tekstuurit. Näiden lisäksi kentän maasto luotiin heightmap – tekstuurin pohjalta. Tämä toteutettiin referenssikohteen maastokartan mukaan.

Kentän objektit luotiin Blender – mallinnussovelluksella. Näihin kuuluivat referenssikohteen aluskasvillisuutta, puita ja kiviä. Mallinnukset toteutettiin modulaarisesti, jossa pienemmistä yksityiskohdista luodaan johdonmukaisella toistolla suurempia kokonaisuuksia. Tällä tavoitteena on vähentää piirtokutsujen ja muistin käytön määrää, joka on hyvin tärkeää kenttäsuunnittelussa, kun kyseessä on monipuolisesta luonnon renderöinnistä reaaliajassa.

Lopuksi luodut objektit sijoitetaan kenttään Cryengineen luodulla kenttäeditorilla opinnäytetyön onnistumisen tarkkailua varten. Maaston muotoileminen uskottavalla tavalla vaatii harkintaa pelikentälle luotujen objektien mukaisesti.

Projekti tarjosi kattavan tietopohjan pelin kenttäsuunnittelusta ja mallintamisesta. Opinnäytetyön tuloksia hyödynnetään indiepelin kehittämisessä ja kasataan portfolioa mahdollisia työtilaisuuksia varten.

### ASIASANAT:

referenssikohde, objekti, tekstuuri, diffuse, specular, normalmap, heightmap, mallintaminen, draw call, renderöinti, Cryengine.

Ilari Laivo

## GAME LEVEL DESIGN

The goal of this thesis is to make a natural looking level in Cryengine with free software. A hill nearby Jyrkkälä, Turku is used as a reference target. The place was chosen because of the good compact environment with a lot of variety in vegetation and terrain. This helps to achieve the goal of the project.

First some reference pictures were taken of the location including some of the initial pictures for texturing. Secondly, the pictures for texturing were made tileable in GIMP photo-editing software. During this project the level uses diffuse, specular and normal mapping textures. Moreover, the level uses a heightmap made out of a map with topological information of the reference location.

The level has objects that are made in Blender modeling software. This includes terrain textures, undergrowth, trees and rocks. The models are made in a modular sense, which helps to use them repeatedly. This is particularly important in natural looking gaming environment where saving draw calls and memory helps to achieve the desired outcome in real time.

In the end the modeled objects are placed into the level in Cryengine with the level editor. The objects are placed into the level with natural look in mind. The success of the project is based on the final look of the level in the game engine.

The project offered extensive information about modeling and level design in games. The results of the thesis are used in indie game development and to gather a portfolio for possible job offers.

### KEYWORDS:

object, texture, diffuse, specular, normalmap, heightmap, modeling, draw calls, rendering, Cryengine.

# SISÄLTÖ

<b>SANASTO</b>	<b>6</b>
<b>1 JOHDANTO</b>	<b>7</b>
<b>2 PELIYMPÄRISTÖN LUOMISESSA KÄYTETYT TEKNIIKAT JA KÄSITTEET</b>	<b>9</b>
<b>3 PROJEKTI</b>	<b>18</b>
3.1 Tavoitteet	18
3.2 Referenssikohde	19
3.3 Työvälineet	20
3.4 Prosessi	21
3.5 Syntyneet tuotokset	23
3.5.1 Maastokartan muuttaminen kolmiulotteiseksi malliksi	23
3.5.2 Tekstuurikuvaus	25
3.5.3 Normal mappien luominen suoraan tekstuureista	28
3.5.4 Kentän objektien mallintaminen ja teksturointi	30
<b>4 JOHTOPÄÄTÖS</b>	<b>37</b>
<b>5 LÄHTEET</b>	<b>40</b>

## KUVAT

Kuva 1. Esimerkki objektista Blenderissä.	11
Kuva 2. Esimerkki tekstuurista.	13
Kuva 3. Esimerkki normal mapista. Kuvassa puun rungon normal map.	15
Kuva 4. Esimerkki kenttäeditorista.	17
Kuva 5. Projektin referenssikohde Turussa.	19
Kuva 6. Vasemmalla referenssikohteen korkeuserokartta, josta oikealla Cryenginelle tehty heightmap.	24
Kuva 7. Yllä esimerkki tekstuurista luodusta normal mapista. Vasemmalla kuvassa normal map –filtterin käyttö yhdellä 4-sample filtterillä. Oikealla useamman tason versio.	28
Kuva 8. Esimerkki maaperään käytetystä tekstuurista.	31
Kuva 9. Esimerkki Unified Detail Mapissa käytettävistä kerroksista.	32
Kuva 10. Vasemmalla on hahmotelma tekstuurista viisinkertaisena tiilirakenteena objektia varten ja oikealla näkyy käytettävä UDM-tekstuuritiedosto.	33
Kuva 11. Aluskasvillisuudessa käytetty alphatekstuuri.	34

Kuva 12. Esimerkki puun tekstuureista.

35

## **TAULUKOT**

Taulukko 1. Käytetyt työvälineet ja ohjelmistot.

20

# SANASTO

Pelimoottori	Pelin käyttämä koodipohja.
Draw call	määritellään graafisen materiaalin käyttöä pelimoottorissa.
Verteksi	Verteksit ovat kolmiulotteisessa tilassa sijaitsevia pisteitä. Näiden avulla on luotavissa objekteja.
Polygoni	Vähintään kolmesta verteksistä muodostuva pinta.
Polygoniverkko/mesh	Polygoneista koostuva geometrinen muoto.
Referenssikuva	Näillä kuvilla määritellään kohteen mallinnettavia yksityiskohtia.
ISO-herkkyys	ISO-herkkyydellä kuvataan valokuvakennolle asetettua herkkyyttä valolle.
Aukko	Aukolla kuvataan kameran objektiivin aukon kokoa.
Objekti	Objektilla tarkoitetaan geometrisistä muodoista luotua kokonaisuutta.
Tekstuuri	Kaksiulotteinen pinta, joka määrittelee objektin pinnan yksityiskohdat.
UV kartoitus	Tekstuuri kartoitetaan objektin pinnalle UV:n avulla. U viittaa pikselin vaakasuoran sijainnin ja V pystysuoran sijainnin.
Normal map	Kaksiulotteinen tekstuuri, joka teeskentelee kolmiulotteista pintaa luomalla tekstuuripinnalle varjostusta.
Specular map	Tekstuuri, joka määrittelee objektin pinnan heijastavuuden.
Gloss map	Tekstuuri, joka määrittelee objektin pinnan kuluneisuuden.
Heightmap	Tekstuuri, jolla määritellään objektin pinnan korkeuseroja.

# 1 JOHDANTO

Pelien kenttäsuunnittelu yhdistää pelinkehityksen osa-alueet kokonaisuudeksi. Kenttäsuunnittelulla on merkittävä rooli pelin toimivuuden kannalta. Hyvätkin elementit voivat kaatua huonoon toteutukseen kenttäsuunnittelun puolella.

Peleissä kentillä tarkoitetaan virtuaalista aluetta, jossa pelaaja toimii saavuttaakseen pelin tavoitteet. Nykyään on varsin yleistä, että nämä alueet toimivat avoimina kokonaisuuksina, jossa pelaaja ei lataa erillisiä kenttiä tavoitteet saavutettuaan, vaan peli jatkuu normaalisti siitä missä pelaaja on samalla, kun peli ilmoittaa tavoitteet saavutetuiksi.

Peligrafiikan mallintaminen vaatii ymmärrystä reaaliaikaisesta renderöinnistä tietokoneilla. Animaatioelokuviissa renderöintiaika on vain hidaste, mutta pelissä tämän täytyy tapahtua sekunnin murto-osissa siten, että pelaajan mahdollisuudet hallita tilannetta pysyy reaaliajassa. Mallintamista ei voi toteuttaa yksityiskohtaisesti, koska tämä lisäisi renderöintiaikaa merkittävästi. Yksinkertaisten mallien päälle on mahdollista luoda kevyesti renderöitäviä pintoja, jotka mukailevat yksityiskohtaista mallia. Kenttäsuunnittelussa tätä tosiasiaa on sovellettava kokonaisvaltaisesti, jotta lopputulos pystytään renderöimään vaaditulla tasolla.

Opinnäytetyössä keskitymme ennalta määrätyn alueen mallintamiseen kentäksi. Kyseessä on Turun Jyrkkälän asutuskeskuksen vieressä sijaitseva mäki. Alue sopii hyvin kuvaamaan kenttäsuunnittelua koskevia graafisia elementtejä, johon tämä opinnäytetyö keskittyy. Näissä graafisissa yksityiskohdissa on pyritty keskittymään kenttäsuunnittelulle tärkeisiin elementteihin ja mietitty niiden soveltuvuutta kentän luomiseen. Kentän graafista materiaalia tuotetaan indie -pelinkehitystä silmällä pitäen siten, että käytössä ovat ilmaistyykalut.

Pelimoottorina opinnäytetyössä käytetään Cryengineä. Pelimoottori eli pelinkehityksessä käytössä oleva koodipohja tarjoaa pelinkehittäjälle työkalut pelin tekemiseen poistaen

tarpeen kehittää näitä alusta pitäen. Nämä työkalut tarjoavat pelille valmiit graafiset elementit, tekoälypohjan, fysiikanmallinnuksen ja tarinankerrontaan käsikirjoituspohjan. Pelissä on myös hiekkalaatikkotila, jossa luotua kenttää voidaan tutkia pelaajahahmon avulla. (Cryengine 2014.)

Projektin tavoitteena on saada kenttä valmiiksi graafisten elementtien osalta ja luoda pohjaa mahdollista portfolioa varten. Lopputuloksessa keskitytään tuotosten luonnollisuuteen. Projektissa käytetään mahdollisimman laajalti Cryengin tarjoamia graafisia ominaisuuksia, jotka ovat mahdollisia indiepelin kehitysympäristön puitteissa ilmaisohjelmilla.

Kenttää varten luodaan pohjakartta, tekstuurit ja objektien mallinnukset. Mallintamiseen käytetään Blenderiä. Blender on avoimeen lähdekoodiin perustuva mallinnus-sovellus (Blender 2014). Tekstuurit luodaan GIMP-kuvankäsittelyohjelmalla, jossa pohjana käytetään järjestelmäkameralla otettuja kuvia.

Opinnäytetyön alussa käydään läpi peliympäristöön sisältyviä tekniikoita, menetelmiä ja selostetaan olennaisimpia termejä. Seuraavassa osassa syvennyttään projektin tavoitteisiin, sisältöön ja graafisen materiaalin mallinnuksen kannalta olennaisiin ratkaisuihin. Lopuksi pohdimme opinnäytetyön onnistumista ja mahdollisia askelia kohti tulevaisuutta.



## 2 PELIYMPÄRISTÖN LUOMISESSA KÄYTETYT TEKNIIKAT JA KÄSITTEET

Tässä luvussa syvennymme projektissa käytettyihin menetelmiin ja peligrafiikan käsitteistöön. Projektin kannalta olennaisia asioita ovat mallinnukseen, teksturointiin ja valokuvaukseen liittyvät termit.

Aluksi kuitenkin käymme läpi millä tavalla kenttäsuunnittelua ylipäättään lähestytään. Esimerkiksi kaupassa käynti on kuvailtavissa seuraavalla tavalla. Kenttinä toimivat koti, lähiö ja kauppa. Tavoitteena toimii siis kaupassa käynti ja lähtöpaikkana toimii koti, josta kuljetaan lähiön poikki kauppaan. Mitä pienempiä kenttiä mallinnetaan, sitä tarkempia yksityiskohtia on syytä mallintaa.

Esimerkissä kodista on syytä mallintaa pienempiäkin esineitä. Tämä on tärkeää pelaajalle esitettävän viestin kannalta. Se kertoo pelaajalle pelattavasta hahmosta ja tarjoaa yksityiskohtia, joita kuka tahansa luonnollisesti tunnistaisi helposti kodistaan.

Lähiön mallintamisessa ei ole varaa keskittyä pienempiin yksityiskohtiin. Tämä toimii alueena, jossa matka taittuu nopeammin ja yksityiskohdat voivat toimia suurempina objekteina. Kohteina, jotka herättävät mielenkiinnon ja edistävät tavoitetta. Yhtenä suurena objektina lähiössä voidaan pitää myös matkan päätöspistettä eli kauppa.

Kaupan ulkomuotoa voidaan pitää osana lähiötä samalla, kun kaupan sisäpuoli muodostaa toisen kentän aiemmin mainitulla tavalla. Kaupassa kaiken ei tarvitse olla interaktiivista, mutta yksityiskohdissa on jälleen syytä mallintaa tarkemmin, ettei hitaasti etenevä katse kaupan hyllyistä huomaa puutteita. Onhan tavoitteena juurikin ostaa ja ennen kaikkea löytää elintarvikkeita, joten näiden mallintaminen on syytä tehdä riittävän tarkasti.

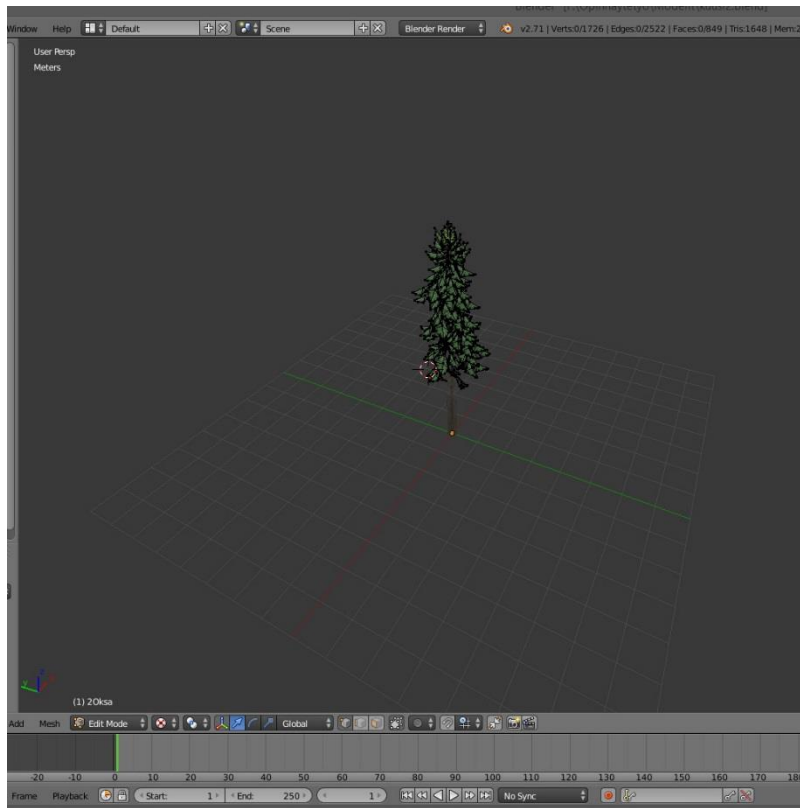
Kuten tästä esimerkistä voimme huomata, kenttäsuunnittelussa on syytä keskittyä pelille tärkeisiin yksityiskohtiin. Tällä tavoin pelaajan on helpompi keskittyä hänelle annettujen tehtävien suorittamiseen.

Seuraavaksi käymme läpi projektissa käytettyjen tekniikkojen ja menetelmien taustaa. **Referenssikuvien** tarkoituksena on tarjota mallintajalle esikuvia mallinnettavista kohteista ja luoda mielikuvaa tavoiteltavasta lopputuloksesta pelikentälle. Kuvien avulla voidaan päätellä tarpeelliset tekstuurit ja mallintamisen kohteet. Myös alueen värimaailmaa ja varjostusta voidaan havainnoida näistä kuvista katsoen. Referenssikuvaa käytetään harkiten ymmärtäen tasapainon löytämistä kompromissien ja välttämättömien yksityiskohtien välillä. (World of Level Design 2011.)

Valon määrä vaikuttaa merkittävästi valokuvan laatuun. Valokuvakennon valoherkkyttä kuvataan **ISO-luvulla** (Wikipedia 2015). Pienin herkkyys on usein ISO100 ja suurin ISO6400. Pienemmällä ISO-herkkyydellä saavutetaan tarkempi kuva, joka sisältää vähemmän kohinaa. Korottaessa ISO-arvoa kohina lisääntyy ja värien tarkkuus heikkenee, mutta valoa ei tarvita yhtä paljon. Valon tarpeen kasvaessa kuvan muodostaminen vaatii pidemmän valotuksen, ellei valoherkkyttä koroteta. Riippuen kuvausolosuhteista tämä voi vaatia kameran käyttöä jalustalla, jotta kuva pysyy terävänä pidemmällä valotuksella.

Toinen kuvan terävyyteen vaikuttava seikka on objektiivin laatu. Tähän voi vaikuttaa säätämällä objektiivin **aukkoa** kameralla. Aukkoa kuvataan F-arvolla. F-arvon osoittaessa pienintä saatavilla olevaa lukua, linssin aukko on tällöin avoimimmillaan. Aukolla pystytään määrittämään terävän alueen laajuutta. Kun kuvaa otettaessa objektiivista käytössä on pienempi aukko, sitä laajemmalta alueelta kuva pysyy terävänä. Aukon koko vaikuttaa myös valovoimaan. Mitä pienempi aukko on käytössä, sitä vähemmän valoa on saatavilla kuvaa varten, joten kuvaa pitää valottaa pidempään. (Wikipedia 2015.)

Referenssikuvien jälkeen peliä varten aloitetaan luomaan graafista materiaalia, jotka sijoitetaan kenttään. Näitä kutsutaan **objekteiksi** (Kuva 1.).



Kuva 1. Esimerkki objektista Blenderissä.

Mallintamisessa objektilla ei tarkoiteta vain yhtä esinettä tai kohdetta. Se voi koostua myös useammasta eri polygoniverkosta. Esimerkiksi aluskasvillisuudessa se voisi tarkoittaa useampaa erillistä kukkaa yhdessä objektissa. Nämä ovat usein sellaisia yksityiskohtia, joiden asettelun luonnollisuutta ei pystytä toistamaan matemaattisesti kevyellä tavalla kenttäeditorilla, vaan ne pitää luoda käsin jo mallinnusvaiheessa. Objektien ominaisuuksiin kuuluvat polygoneista koostuva polygoniverkko ja tekstuuripinta (Cryengine 2014).

Polygon mesh eli **polygoniverkko** on kahdesta tai useammasta polygonista koostuva kolmiulotteinen objekti. Käytämme raportissa jatkossa polygoniverkosta mallinnuksessa yleisesti käytettyä termiä **mesh**. (Palamar 2012, 110.)

**Polygoni** eli monikulmio on kolmiulotteinen suljettu muoto, joka muodostuu vähintään kolmesta verteksistä (vertex). Verteksi osoittaa pisteen kolmiulotteisessa tilassa XYZ-koordinaattien mukaan. Kaksi viivalla yhdistettyä verteksiä muodostaa polygonin reunan (edge). Kun reunat sulkeutuvat yhdeksi polygoniksi kolmesta tai useammasta verteksistä, saamme sille pinnan (face). (Palamar 2012, 110-112.)

Useimmiten polygoni muodostuu tasaisista nelikulmioista, koska tällä tavoin objektille on luotavissa käytännöllisiä tekstuureja, joiden toistaminen kolmiulotteisessa ympäristössä ei vaadi näytönohjaimelta merkittävää muistin käyttöä. Nelikulmio mahdollistaa myös meshin jakamisen pienempiin polygoneihin ilman muodon hajoamista laskennallisiin virhearvioihin monimutkaisemmasta meshistä. Tämä näkyy tekstuurin venymisenä polygoniverkon pinnalla. Näistä menetelmistä kerrotaan lisää kohdassa teksturointi. Polygonilla voi olla enemmän kuin neljä kulmaa. Näitä kutsutaan N-goneiksi n-kirjaimen viittaessa kulmien lukumäärään (Ward 2008, 6).

Referenssikuvien lisäksi tässä opinnäytetyössä valokuvista tehdään myös **tekstuureja**. Tekstuurilla määritellään objektien ja maaston pintojen kaksiulotteisia yksityiskohtia. Tekstuuri koostuu kaksiulotteisesta neliöstä (Kuva 2.). Niiden tarkkuus vaihtelee yleensä 128x128 pikselin ja 4096x4096 pikselin välillä. Normaalisti tekstuurilta vaaditaan reunojen saumattomuutta ja neutraaliutta. Tekstuurista ei saa nousta esiin yksityiskohtia, jotka korostuisivat toistuvuutena objektien pinnoilla.



Kuva 2. Esimerkki tekstuurista.

Meshin pinta sisältää tekstuurikoordinaatteja, jotka ovat talletettuina vertekseissä. Näiden koordinaattien avulla pystymme lisäämään niille haluttuja tekstuureja. Näitä kutsutaan myös nimellä **UV**. Kirjaimet U ja V osoittavat pikselin sijainnin tekstuurissa. U kertoo pikselin sijainnin vaakasuorassa, kun V kertoo pystysuorassa. (Palamar 2012, 596.)

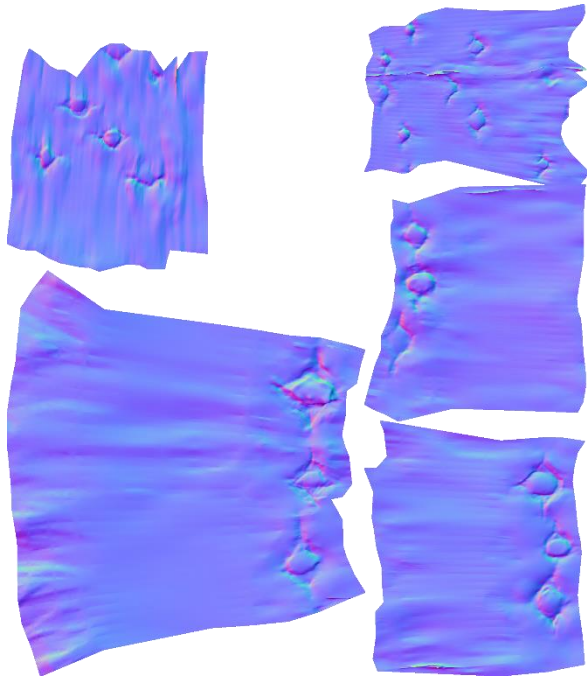
Tekstuureilla on merkittävä rooli objektin ulkonäön kannalta. Peleissä tekstuureille on tarjolla monta erilaista tekniikkaa, joilla on mahdollista parantaa yksityiskohtien monipuolisuutta ilman merkittävää tehonkulutusta. Projektissa käytetään viittä eri tyyppistä tekstuuria. Ne ovat diffuse map, specular map, gloss map, normal map ja heightmap.

Tekstuurit määrittelevät valon käyttäytymisen objektien pinnoilla. Objektin pinta heijastaa kahdenlaista väriä. Objektille ominaista väriä nimeltä diffuusi (diffuse color) ja siihen osoitettavaa valon väriä (specular color). Se mitä väriä ihmissilmä näkee objektin pinnalla, perustuu objektin ominaisuuksiin. Esimerkiksi metalli heijastaa enemmän siihen osoitettavaa valoa, kuin sille ominaista väriä. Puun pinnasta näemme selkeämmin

objektille ominaiset värit. Näitä kahta elementtiä käytetään pohjatekstuureina, joiden päälle lisätään muita yksityiskohtia. Nämä ovat nimeltään **diffuse map** ja **specular map**. (Toledo 2010.)

Pelikentän värit syntyvät valon väristä ja objektin väristä. Voidaan puhua myös pinnan heijastavuudesta, joka johtuu materiaalien yhdistelmästä tai pinnan kuluneisuudesta, jonka ansiosta siitä voi nähdä heijastuksen. Esimerkiksi peili tai puulattia, joka on kulunut sileäksi. Realistisen heijastuksen saavuttaminen vaatii paljon laskentatehoa, joten sitä ei peleissä vielä käytetä. On kuitenkin mahdollista luoda uskottavan näköinen harha heijastavasta pinnasta käyttämällä yksinkertaisia tekstuurikerroksia, kuten specular mappia ja cube mappia. Cube mapilla tarkoitetaan kuvaa, joka heijastuu objektin pinnasta sille osoitetulla alueella. Cryenginellä meillä on käytettävissä tämän lisäksi **gloss map**, joka määrittelee pintojen kuluvuuden (Cryengine 2014).

**Normal mapilla** luodaan objektille kaksiulotteinen pinta, joka kauempaa katsottuna imitoi kolmiulotteista pintaa (Kuva 3.). Se on yleisin tekstuuripinta kolmiulotteisten pintojen imitoimiseen keveytensä vuoksi. Koska kyseessä on kevyt tekstuuripinta, se ei lisää varsinaisia korkeuseroja. Tämän voi havaita tarkastelemalla tekstuuria läheltä, jolloin kolmiulotteinen vaikutelma tietyistä kulmista katsottuna katoaa. Tämän huomaa etenkin siluetista.



Kuva 3. Esimerkki normal mapista. Kuvassa puun rungon normal map.

Normal mapissa jokainen pikseli sisältää normaalin. Se on vektori, joka kertoo pikselin kaltevuuden tekstuuripinnalla, jonka pohjalta varjostus syntyy. Jotta kolmiulotteinen vaikutelma saataisiin aikaan kaksiulotteisella tekstuuripinnalla, pitää sille osoittaa pinnan kaltevuussuunta ja syvyys. Tämä toimii geometriasta tutuilla XYZ-koordinaateilla, jossa tekstuuripinnalla käytetään värejä osoittamaan suuntaa ja syvyyttä. Väreinä toimivat punainen, vihreä ja sininen. Punainen osoittaa pinnan kaltevuuden X-suunnassa. Vihreä osoittaa pinnan kaltevuuden Y-suunnassa. Sininen ilmoittaa pinnan syvyyden Z-suunnassa.

Normal mappia voidaan käyttää kahdessa tilassa. Vaihtoehtoina toimivat tangenttitila (tangent-space) tai objektitila (object-space). Peleissä suositaan tangenttitilaa sen paremman soveltuvuuden vuoksi. Tangenttitilan normal mapit käyttävät verteksitietoa nimeltä tangentin perusta (tangent basis). Tämä antaa tiedon normal mapin kaltevuudesta valon lähteeseen nähden.

Tangent basis –tiedon avulla normal map voi kommunikoida maailman tilan (world-space) tarjoaman valon kanssa. Valon suuntaa verrataan pikselin kaltevuuteen ja ne valaistaan sen mukaisesti. Tämä tapahtuu kääntämällä maailman tilan valo tangenttitilaan tai päinvastoin, riippuen varjostuksessa (shader) käytettävästä lähestymistavasta (Ward 2008, 118-121).

Pelissä edellä mainitut tekstuurit ja objektit sisältävät piirtokutsun. **Draw call** eli piirtokutsu tulee pelin graafisista elementeistä. Erikoisefektit, käyttöliittymän ulkonäkö, valot ja varjot tarvitsevat myös omat kutsunsa. Jokainen erillinen kutsu hidastaa lopullisen renderöinnin tapahtumista (Cryengine 2014).

Piirtokutsuja voi vähentää Cryenginissä unified detail mappingillä. **Unified detail map** on Cryenginissä käytetty teksturointitekniikka, joka yhdistää usean tekstuurin yhteen tiedostoon. Tekstuuritiedosto sisältää objektin ambient occlusionin, gloss mapin (specular map) ja normal mapin (Cryengine 2013). Tähän tekniikkaan syvennyttään tarkemmin tuotos-osiossa.

Kun objektit on mallinnettu ja niille on osoitettu asianmukaiset tekstuurit, voimme siirtää ne kenttäeditoriin. **Kenttäeditori** on pelimoottorille luotu graafinen työkalu (Kuva 4.), jolla pyritään helpottamaan pelikenttien luontia pelimoottorin ulkopuolella luodun materiaalin pohjalta.





## 3 PROJEKTI

### 3.1 Tavoitteet

Projektin tavoitteena on luoda pohjaa kenttäsuunnitteluun FPS indiepelille. FPS on First Person Shooter, jossa pelaajahahmoa ohjataan pelaajan silmien näkökulmasta. Tavoitteeseen pyritään käyttämällä avoimeen lähdekoodiin pohjautuvia ja ilmaisia työkaluja. Näihin kuuluvat Blender, MeshLab, GIMP ja xNormal -sovellukset. Graafisen lopputuloksen toteuttaminen tapahtuu Cryengininen avulla. Lopputuloksesta pyritään luomaan portfolioon soveltuvaa materiaalia, joka tämän opinnäytetyön puitteissa keskittyy luonnollisuuteen.

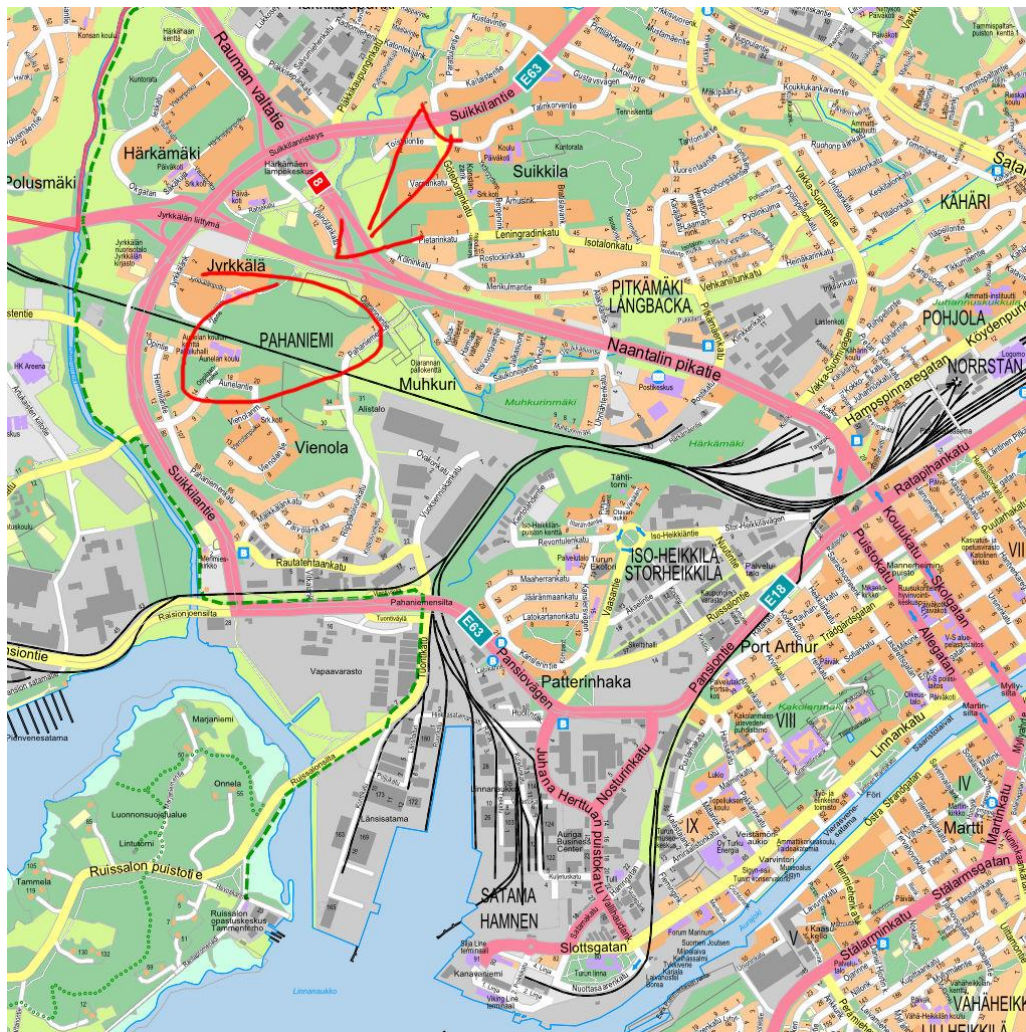
Projektia varten kuvataan tekstuurikuvia. Kuvia kehitetään alustavasti Lightroomilla, jolla tekstuurikuvista muokataan tekstuurikäyttöön sopivia. Kuvista poistetaan kameran linssien aiheuttamat vääristymät, jonka jälkeen ne tallennetaan pakkaamattomassa muodossa GIMP:n lopullista kuvankäsittelyä varten.

Kuvankäsittelyyn käytetään GIMP:iä. GIMP:llä viimeistellään objektien tekstuurit. xNormal -sovellusta käytetään normal mappien tekemiseen Blenderillä luoduista yksityiskohtaisista mallinnuksista.

Mallinnussovelluksena käytetään Blenderiä, jonka kautta mallinnuksista tehdään yksityiskohtaiset versiot. MeshLabia käytetään mallinnusten polygonien vähentämiseen **low poly -mallia** varten, joka siirretään Blenderin kautta Cryengineen. Low poly –malli tarkoittaa vähennetyn polygonimäärän muodostamaa objektia. Tämä onnistuisi myös Blenderissä, mutta polygonien vähentäminen MeshLabin avulla tuottaa laadukkaamman ja kontrolloidumman lopputuloksen. Jokainen low poly –malli käyttää pehmeäpintaista varjostusta tässä opinnäytetyössä.

### 3.2 Referenssikohde

Kenttäsuunnittelun demonstraatiota varten on valittu referenssikohdeksi Turun esikaupunkialueella sijaitseva paikka nimeltä Pahanieniemi (Kuva 5.). Alue kattaa hyvin Etelä-Suomessa yleisesti esiintyvän käenkaali-mustikkatyypisen metsän kasvillisuutta. Mäen reunoilla kasvaa pääasiassa kuusi- ja lehtipuita ja niille tyypillistä aluskasvillisuutta. Kallion päällä ja Jyrkkälästä lähtevän polun varrella mäntypuu on yleisin puulaji. Avokallion vuoksi alueella on myös sille tyypillisempää aluskasvillisuutta. Alueen pienen koon ja maaston monipuolisuuden vuoksi paikasta on löydettävissä hyviä referenssikohteita, joiden avulla on mahdollista demonstroida opinnäytetyön aihetta.



Kuva 5. Projektin referenssikohde Turussa.

Lopputuloksessa tavoitellaan Cryengin dokumentaation suositusten noudattamista piirtokutsujen ja polygonimäärien suhteen. Tämä tapahtuu suuntaa-antavasti, koska grafiikan laadun määrittely suoritetaan vasta pelinkehityksen loppuvaiheessa.

### 3.3 Työvälineet

Käytössä olevat laitteet (Taulukko 1.) ovat olleet henkilökohtaisessa käytössä jo ennen opinnäytetyön alkua. Opinnäytetyön sisältö ei ole vaikuttanut niiden valintaan.

<b>Kuvauslaitteisto:</b>	<b>Tietokone:</b>	<b>Ohjelmisto:</b>
Canon 60D	Intel Core i5 2500k	Gimp 2.8
18-55mm IS II -objektiivi	8GB muistia	Blender 2.71
Canon RC-6 Langaton etälaukaisin	128GB SSD-kovalevy	Adobe Lightroom 5
Slik Monopod 350	500w virtalähde	Windows 8.1
	Nvidia Geforce GTX660ti	Cryengine 3.5.8
	2GB Näytönohjain	MeshLab 1.3.8 BETA
		xNormal v.3.18.10

Taulukko 1. Käytetyt työvälineet ja ohjelmistot.

Mallintamiseen laitteisto tarjoaa riittävästi tehoja, eikä työskentely vaikuta kankealta. Rajat tulevat vastaan vain joissakin objektien renderöintitapauksissa, joissa polygonimäärät nousevat poikkeuksellisen suuriksi. 8 gigatavua muistia on riittävä määrä

käyttötarkoituksiin nähden. Tietokoneen prosessori on edelleen tehojen puolesta riittävän hyvä pelikäyttöön, joka toimii sopivana mittarina pelinkehityksessä (Tom's Hardware 2015). Se riittää myös tehokkaampien näytönohjainten tarpeisiin, jotka vaativat enemmän laskentatehoa tiedon siirtämiseen näytönohjaimelle. SSD-kovalevy pitää huolen siitä, että työskentely on sulavaa eri sovellusten välillä alusta loppuun.

Tietokoneen näytönohjaimena toimii Nvidian GTX660ti 2 Gigatavun muistilla. Näytönohjain julkaistiin vuonna 2012, joten suorituskky ei vastaa aivan tehokkaimpia tarjolla olevia näytönohjaimia (Tom's Hardware 2015). Tämä kuitenkin riittää opinnäytetyön tarpeisiin.

Käytössä oleva kameralaitteisto on vakavalle harrastajalle suunnattu järjestelmäkamera. Käytössä on kittilinsi, joka tarkoittaa kameran mukana tullutta objektiivia. Nämä eivät vastaa laadullisesti ammattikäyttöön tarkoitettuja laitteita, mutta mahdollistavat vastaavanlaisen kontrollin kuvan ottamisen yhteydessä.

### 3.4 Prosessi

Projekti aloitetaan referenssikuvauksella. Kuvat analysoidaan ja sen jälkeen siirrytään tekstuurikuvauksessa otettujen kuvien viimeistelyyn Adobe Lightroom 5 ja GIMP 2.8 - kuvankäsittelysovelluksilla. Adobe Lightroom on Adoben kehittämä photoshopista kevyempi ja tehokkaampi työkalu vain valokuvia varten. Kyseessä on myös ainoa sovellus tässä projektissa, jolla on maksullinen lisenssi. Tämä helpottaa työskentelyä raakakuvien editointivaiheessa, mutta ei ole välttämätön työkalu seuraavaan vaiheeseen siirryttäessä. Lightroomiin on myös lisätty linssikohtaisia esiasetuksia, joilla poistetaan vinjetointia, värivirheitä ja linssivääristymiä. Tällä esiasetuksella kuvasta saadaan tasainen ja väreiltään toimiva lopputulos. Lightroomissa viimeisenä toimenpiteenä kuvasta leikataan ennaltamäärätyn kokoinen palanen kuvasta tarkennettuun osaan tekstuurin viimeistelyä varten, joka tehdään GIMP:ssä.

GIMP on GNU-lisenssiin perustuva avoimen lähdekoodin kuvankäsittelysovellus. GIMP-sovelluksessa tekstuuri viimeistellään tiilimäiseksi kokonaisuudeksi, jossa tekstuurissa

toistuvat yksityiskohdat eivät korostu. Ensimmäiseksi poistetaan tekstuurin reunat Offset –työkalulla, jossa leikatun tekstuurikuvan jokainen kulma sijoitetaan kuvan keskelle. Tämän jälkeen tunnistettavat reunat korvataan Kloonit-työkalulla siihen sopivilla yksityiskohdilla, joilla nämä reunat häivytetään kuvasta kokonaan. Jos kuvassa esiintyy edelleen toistuvia kohtia, poistetaan nämä korvaamalla ne neutraaleimmilla yksityiskohdilla.

Seuraavaksi siirrytään objektien mallintamiseen. Tähän tarkoitukseen käytössä on Blenderin versio 2.71. Blender on GNU-lisenssiin perustuva avoimen lähdekoodin mallintamistyökalu. Blenderille on tarjolla siirtotyökalu, jonka avulla on mahdollista siirtää objekteja pelimoottorin ymmärrettävässä muodossa. Tämän työkalun nimi on CryBlend. Alunperin yhden ihmisen kehittämä työkalu on saanut taakseen pienen kehittäjätiimin, joka pyrkii kehittämään lähestyttävän työkalun mallinnusten tuomiseen Blenderistä Cryengineen. Dokumentointi on vielä tässä vaiheessa hyvin keskeneräistä, joten monet mahdolliset ongelmatilanteet pitää selvittää itse.

Viimeisenä vaiheena projektissa luodut objektit asetetaan ennaltamäärätyyn kenttään tavoitteiden mukaisesti. Tämä tapahtuu Cryengin Sandbox-editorilla. Cryengine -pelimoottorilla saa kehittää peliään ilmaiseksi, jos sitä tehdään aluksi ei-kaupallisena työnä. Kaupalliseen käyttöön siirryttäessä peliä varten pitää hankkia kuukausimaksullinen lisenssi. Tämä helpottaa indiepelinkehityksen alkuun pääsemisessä, kun pelimoottoriin ei joudu tekemään merkittävää alkusijoitusta.



### 3.5 Syntyneet tuotokset

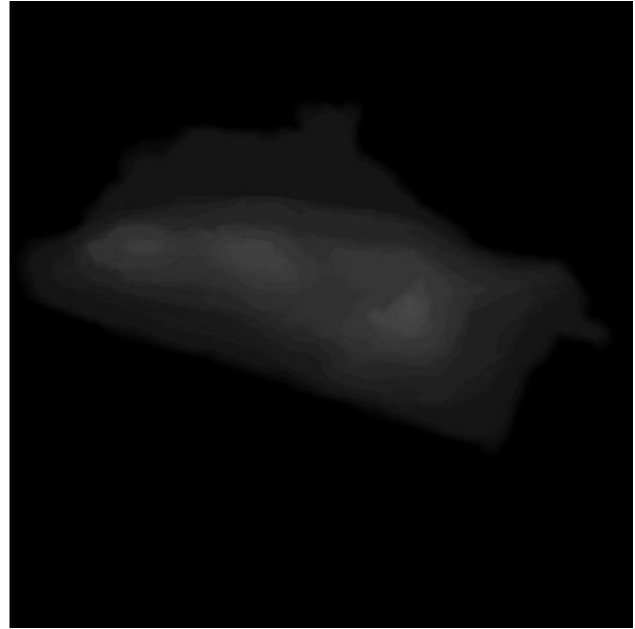
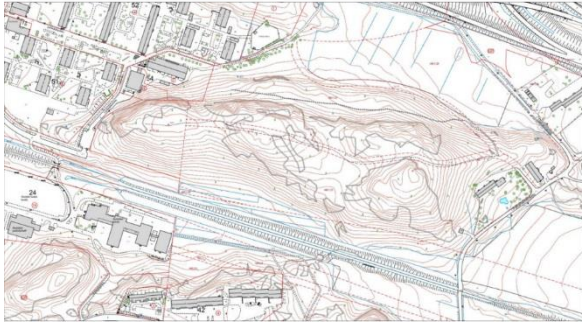
Tuotoksissa esitellään projektissa käytettyjä ratkaisuja, jotka ilmaissovellusten käytössä poikkeavat normaalista ohjeistuksesta Cryengin dokumentoinnissa. Tämän lisäksi osiossa avataan projektin tavoitteen mukaisia ratkaisuja objektien mallinnusta ajatellen.

Aluksi käydään läpi referenssikohteen pohjakartan mallintamista korkeuserojen perusteella. Sen jälkeen keskitytään tekstuurikuvausta koskeviin yksityiskohtiin ja kameralaitteiston asetuksiin. Tekstuurikuvauksen jälkeen kuvaillaan diffuse-tekstuuriin pohjautuvaa normal map –teksturointitekniikkaa, joka on käytössä kaikissa kiinteissä polygonipinnoissa. Lopuksi käydään läpi projektissa käytettyjä maaston, kivien, aluskasvillisuuden ja puuston teksturointi- ja mallinnusmenetelmiä.

#### 3.5.1 Maastokartan muuttaminen kolmiulotteiseksi malliksi

Kentän mittoina käytetään 1024x1024 pikselin karttapohjaa, jossa jokainen pikseli vastaa yhtä metriä pelissä. Tämä ei ole tarkka suhdeluku verrattuna kartasta otettuun alueeseen, vaikka sekin olisi mahdollista. Näin säästetään aikaa, eikä kyseessä ole olennainen seikka opinnäytetyön tavoitteita ajatellen.

Paikkaa ympäröi meri, jotta rajausta olisi selkeää. Selkeämpi mukaelma alueen muusta ympäristöstä saattaa tulla projektiin myöhemmässä vaiheessa, mutta sitä ei sisällytetä opinnäytetyöhön.



Kuva 6. Vasemmalla referenssikohteen korkeuserokartta, josta oikealla Cryenginelle tehty heightmap.

Cryengine karttapohja on nimeltään **heightmap** (Kuva 6.), joka on nimensä mukaisesti maaston korkeuskartta, josta erottuvat korkeuserot. Korkeuserot muodostuvat harmaasävyistä, jossa vaaleampi alue on korkeammalla. Tämän lisäksi heightmappia on mahdollista käyttää myös muuhun tarkoitukseen objektien yksityiskohtien lisäämiseksi vastaavalla tavalla.

Kentän luomiseen käytettiin maastokarttaa, jossa mainitut korkeuserot muutettiin harmaasävyiksi. Lopputulos saatiin aikaan päättämällä alin harmaasävyyn arvo maastokartan alimmaksi kohdaksi ja korkein kohta korkeimmaksi arvoksi.

Maastokartta piirrettiin harmaasävyiseksi GIMP kuvankäsittelyohjelmalla käyttämällä retussiruisu –työkalua. Tällä pyritään saamaan eri sävyjen välille pehmeyttä, jotta lopputulos ei näyttäisi pelimoottorissa liian palikkamaiselta. Kenttä piirrettiin kerroksittain alimmasta korkeuserosta lähtien. Alimmaksi kohdaksi maastokartasta on valittu 10 metriä, josta myös kentän rajaus tulee alueen pohjoispuolella. Etelässä rajauksena toimii junarata. Idässä ja lännessä asutusalueet. Harmaasävyjen kerrokset piirrettiin 5 metrin



välein, joten tarkka korkein kohta jää lähinnä näennäiseksi. Tämä kuitenkin riitti hyvin mukailemaan aluetta vastaavaa maastoa.

Lopputulos ei aluksi näyttänyt hyvältä kenttäeditorin puolella. Harmaasävyjen korkeuserot olivat aivan liian radikaaleja ja lopputulosta piti korjata vähentämällä kontrastia ja himmentämällä alkuperäistä kuvaa, jotta se vastaisi paremmin maastokartan korkeuseroja. Pohjakartta oli myös edelleen hyvin palikkamainen mahdollisesti käytetyn tiedostomuodon vuoksi, koska GIMP ei tue 16-bittisiä kuvatiedostoja. Karttapohjan tiedostona jouduttiin siten käyttämään 8-bittistä bmp-tiedostomuotoa.

Onneksi lopputulosta voitiin hienosäätää myös kenttäeditorin puolella. Palikkamaisuuden sai eliminoidua kenttäeditorin maaston "smooth"-toiminnolla. Samalla myös pelimoottorin tulkintoja alkuperäisen tiedoston korkeuseroista pystyttiin korjaamaan "set terrain max height" -asetuksella. Tässä käytettiin logiikkaa, joka vastasi 8-bittisen bmp-tiedoston harmaasävyjä, joita sillä on 256. Viimeisenä silauksena meren pintaa nostettiin lähemmäksi alinta kohtaa, joka oli selvästi harmaasävyyn 0 arvoa korkeammalla. Näin tehtiin siksi, koska harmaasävyillä piirtäminen ei onnistu lähellä nollaa olevilla arvoilla, jotka olisivat luoneet hyvin vaativat olosuhteet niiden editointiin.

Alkuperäistä tiedostoa työstettiin 40-160 arvoisten harmaasävyjen välillä eli jokainen metri vastasi neljää harmaasävyä sillä oletuksella, että korkein kohta sijaitsee 40 metrissä. Pyöristykseen päädyttiin 5 metrin välein piirrettyjen harmaasävyjen pohjalta.

### 3.5.2 Tekstuurikuvaus

Tekstuurikuvauksen kohteena ovat referenssikuvauksessa käytetyt kohteet ja yleensä alustavat kuvat tekstuureja varten ovat otettu samalla kertaa. Referenssikuvien avulla on mahdollista syventyä kohteelle ominaisiin yksityiskohtiin ja miettiä ympäristössä olennaisia tekstuureja.

Kuvauksessa on tärkeää huomioida tekstuureille tärkeitä elementtejä. Siinä ei saa olla sellaisia yksityiskohtia, jotka näkyisivät lopputuloksessa ilmeisenä toistuvuutena. Peleissä on käytettävissä vain rajallinen määrä muistia, joten esimerkiksi yhden puun kohdalla ei voida käyttää lukuisia tekstuureja yksityiskohtia korostamaan. Tähän lopputulokseen päästään tekstuuria viimeistelemällä kuvankäsittelyohjelmalla.

Tekstuurikuvauksen tärkeimpiä kriteerejä ovat kuvan tasaisuus ja laatu. Ihanteellinen tekstuurikuva on otettu pilvisessä säässä pienellä aukolla ja pienimmällä mahdollisella ISO-herkkyydellä. Kuva on otettu kohtisuoraan tasaisesta pinnasta (ellei kyseessä ole poikkeuksellinen yksityiskohta), eikä siinä näy valon luomia korostumia, johon yleensä pilvinen sää riittää. Tällä pidetään huoli siitä, että mahdollisimman suuri alue kuvan pinta-alasta on käytettävissä tekstuurissa. Kuvassa korostetaan tekstuurimaisuutta, Esimerkiksi puun kaarnan juonteita, pohjaväriä ja pinnan kulumista.

Asetuksissa pyritään mahdollisimman alhaiseen kohinatasoon ja kuvan vääristymään, jotta yksityiskohdat säilyisivät hyvänä ja tasaisena koko kuvassa. Kameran valinnassa ei ole erityistä syytä. Mutta kyseisen kameran käyttö on ennestään tuttua ja helpottaa tekstuurikuvaukseen kohdistuvaa pohdintaa. Kennon koolla voisi vaikuttaa kuvanlaatuun. Ammattilaiskäytössä olevalla kennolla voisi saada vähemmän kohinaa kuvaan, myös polttoväliä saadaan pienemmäksi. Tosin tekstuurikuvan ottaminen onnistuu myös hyvänlaatuisella kameralla älypuhelimessakin, jonka eduksi voisi kutsua pientä polttoväliä. Tämä helpottaisi tasaisen kuvan saavuttamisessa, joka tekstuurikuvissa on tärkeää.

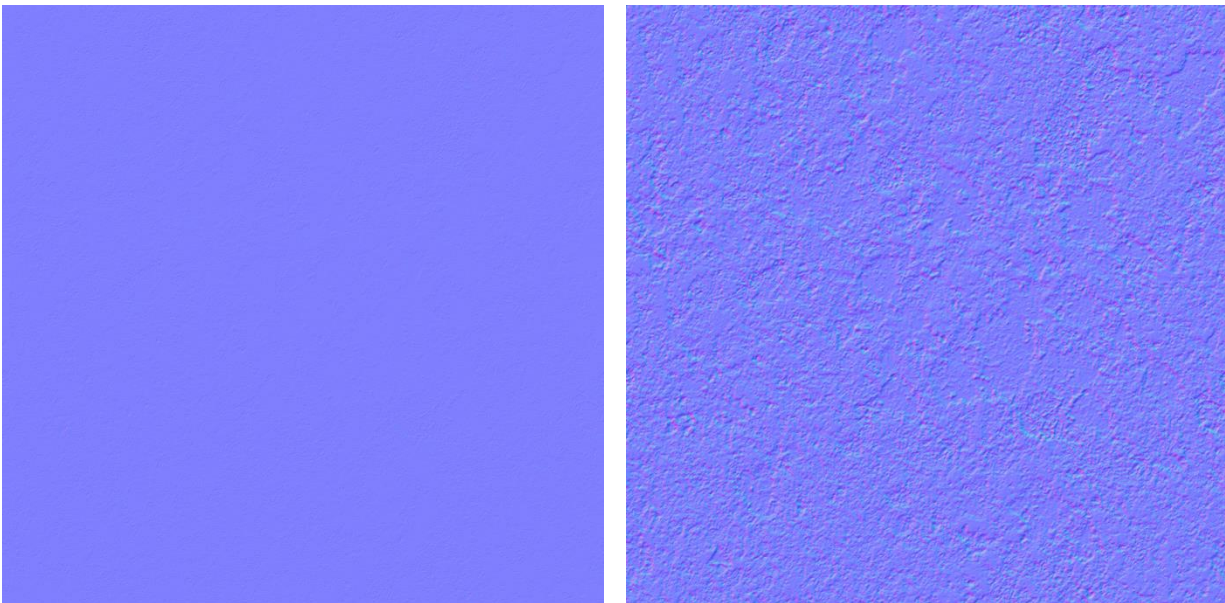
Koska kyseessä ei ole ammattilaistason kennosta, joudutaan tekemään kompromisseja valon määrän suhteen. Kuvauksessa käytetään 5.6 kokoista aukkoa, jonka pitäisi riittää tasaiseen ja terävään kuvaan (Photozone 2007). Tämän pitäisi poistaa myös kuvassa esiintyvää vinjetointia eli kuvan reunojen tummumista. On kuitenkin huomioitava, ettei kuva tule tekstuurikäytössä kokonaan käyttöön, joten pienet vääristymät eivät aiheuta ongelmia kuvan reuna-alueilla. Nämä leikataan pois lopullisesta tekstuurista joka tapauksessa.

ISO-arvona pyritään käyttämään ISO800 ja siitä alempia valoherkkyysä. Projektin alkumetreillä tuli kuitenkin selväksi, että pilviset olosuhteet metsässä kuvattaessa ovat haastavia ja tilanteesta riippuen ei voida päästä tähän tavoitteeseen. Tämä on kuitenkin suurin valoherkkyys, jolla kuvan kohinataso pysyy hyvänä (DxOMark 2014).

Tekstuurikuvauksessa käytetään 18mm:n polttoväliä, joka on käytetyn objektiivin pienin polttoväli. Tämän avulla kuva pysyy mahdollisimman tasaisena ja tarkkana käytettävässä tekstuurissa. Kuvauksen aikana kuitenkin huomattiin esimerkiksi puun rungon olevan liian kaareva tähän tarkoitukseen, joten tarkennettu pinta-ala pysyi rajoittuneena. Alustavien tekstuurikuvausten pohjalta voidaan kuitenkin todeta, että 18mm osoittautui ainoaksi varmaksi valinnaksi hyvänlaatuisille tekstuureille.

### 3.5.3 Normal mappien luominen suoraan tekstuureista

Projektin aikana on käytössä kahdenlaista normal mappia. Toisessa käytössä on Unified Detail Mapping –tekniikka, jonka normal map-pinnalla keskitytään tämän projektin aikana suurempien yksityiskohtien varjostamiseen. Tämän menetelmän käyttöönottoon liittyvistä yksityiskohdista puhutaan tarkemmin kiven mallintamisen yhteydessä. Toinen näistä keskittyy pintojen yksityiskohtiin omana tekstuuritiedostonaan. Tälle on pääasiassa käytössä kuvista luotuja tekstuureja, jotka vaativat vielä erillisen tekstuuripinnan varjostusta varten. Näissä käytetään hyväksi diffuse-tekstuuria, jotka muutetaan normal mapiksi GIMPin avulla. Kyseessä ovat sellaiset pienet yksityiskohdat, joiden toteuttaminen yksityiskohtien mukaisesti olisi käsin liian vaikeaa ja työlästä. Näihin normal mappeihin lukeutuvat tässä opinnäytetyössä maaperä, puun runko ja kallion pintatekstuuri.



Kuva 7. Yllä esimerkki tekstuurista luodusta normal mapista. Vasemmalla kuvassa normal map –filtterin käyttö yhdellä 4-sample filtterillä. Oikealla useamman tason versio.

Normaalisti GIMP:n normalmap-suodatin luo heikon mukaelman tekstuurin kolmiulotteisesta pintarakenteesta (Kuva 7.). Vahvistaaksemme tätä pintarakenteen kolmiulotteisuutta käytämme hyväksi CG Texturesin tutoriaalia kuvatekstuurin kolmiulotteisuuden palauttamisesta. Alkuperäisessä tutoriaalissa käytössä on Adobe Photoshop. Tekniikkaa on sovellettu GIMP kuvankäsittelyohjelmaan.

Aluksi kuva muutetaan mustavalkoiseksi. Tämän jälkeen se viedään normalmap -filtterin läpi 4 sample -asetuksella. Kuvaan saadaan parempi kolmiulotteinen vaikutelma, kun se kopioidaan toiselle tasolle ja tämä muutetaan overlay:ksi. Näin lopputuloksessa korostuu yksityiskohtien kolmiulotteisuus, koska overlay vahvistaa aiempien tasojen luomaa näkyvää kuvaa. Jotta overlayn vaikutukset näkyisivät lopputuloksessa, pitää tämän olla tasojen ylimpänä.

Seuraavaksi kopioidaan uudelle tasolle aiemmin luotu kopio overlaystä. GIMP luo kopion automaattisesti alkuperäisen tason päälle, joten normal mappia tehdessä ei tarvitse erikseen siirtää näitä luotuja tasoja ylimmäksi, johon ne pitää sijoittaa.

Kolmiulotteinen vaikutelma saisi tällä menetelmällä luultavasti liian karheen lopputuloksen, joten jokainen kerros pehmennetään tekstuurin tarpeen mukaan. Menetelmää on muutettu alkuperäiseen ohjeistukseen nähden siten, että jokaisessa tekstuurissa on käytössä 32 kerrosta, mutta kerrokset määrittyvät kontrastin mukaan GIMP:n -127 ja 127 arvon välillä. Tästä tulee valkotasapainon 255 harmaasävyä, joiden pohjalta saavutamme kolmiulotteisen muodon pinnan valoisuuden mukaan. Pinta on tasaisempi alkuperäisen tutoriaalin lopputulokseen nähden, mutta sisältää tarkemman pintatiedon tekstuuriin nähden.

Aivan lopuksi normal map pitää normalisoida, koska kuvan vektorin väriarvo ei enää vastaa vaadittua yhden arvoa (CG Textures 2007).

### 3.5.4 Kentän objektien mallintaminen ja teksturointi

Tässä kappaleessa kuvataan aluskasvillisuuden, maaston ja puiden luontia niiden eri vaiheissa. Jokaisesta osa-alueesta tarjotaan esimerkki, jonka avulla kuvataan miten näistä saadaan luonnollisen näköisiä ja minkälaisia tekstuureja näitä mallinnuksia varten on tehty.

Jokaista kasvia kohden on tutkittu niiden kasvutapoja ja pohdittu yleisiä johdonmukaisuuksia, joita voimme toistaa kaikissa mallinuksissa. Tällä tavoin materiaaleista pystytään luomaan monipuolisia ja uskottavan näköisiä.

Projektin aikana digitaalista veistämistä käytetään tiettyjen objektien muotoiluun. Digitaalinen veistäminen muistuttaa savityön tekemistä, joten se on erittäin oivallinen ja nopea tapa työstää luonnonmukaisia malleja. Tämän projektin aikana näihin kuuluvat kallion ja puun rungon mallit. Työ tehdään Blenderillä dynaamisella topologialla, joka lisää verteksejä työstettävään objektiin sen perusteella, jos lisäämäsi yksityiskohdat eivät onnistu ilman artefakteja (Blender Wiki 2015).

Valmis veistos siirretään .ply –tiedostomuodossa Meshlab-työkaluun, jossa polygonien määrää lasketaan alle 2000 polygoniin mallin käyttötarkoituksen mukaan. Työkalussa valitaan 'Quadric Edge Collapse', jolla voidaan määritellä tarkkaan pienennetyn polygonimäärän vaikutukset malliin (Shapeways 2015).

Veistetyistä mallista luodaan normal map xNormal –baketyökalulla sen jälkeen, kun mallille on määritetty UV-kartta pienennetyn polygoniverkon päälle Blenderissä.

**Maaperää** varten luodaan toistuva tekstuuri, jonka reunoja ei ole erotettavissa. Tekstuurin ideana on luoda mielikuva maaperän tyypistä (Kuva 8.). Tekstuurissa ei saa olla erottuvia yksityiskohtia, koska tämä poistaisi toistuessaan tekstuurin luonnonmukaisuuden. Projektin maaperätekstuurit ovat 1k-resoluutioisia ja niissä käytetään normal mappia, joka on luotu GIMP:n kerrostettua normalmap-suodatusta hyväksikäyttäen.



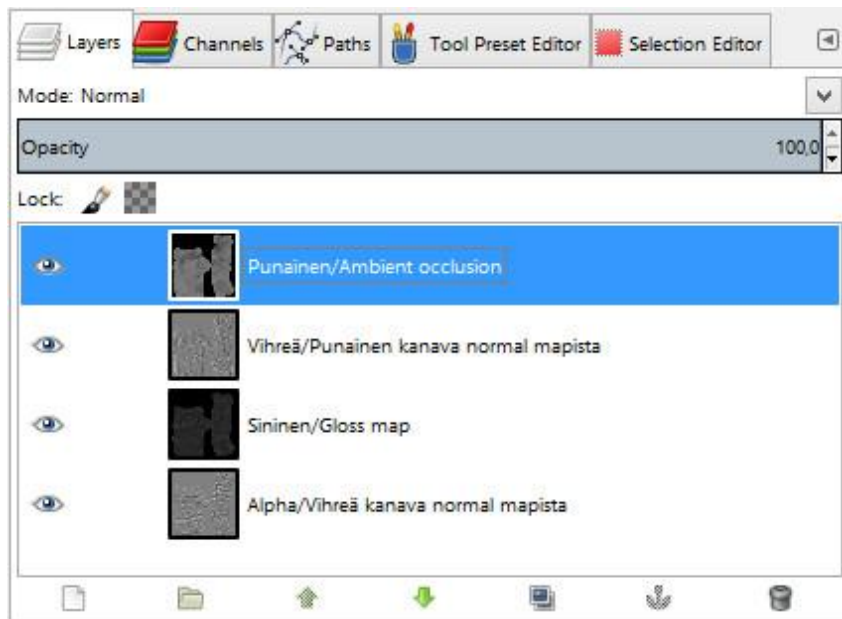
Kuva 8. Esimerkki maaperään käytetystä tekstuurista.

Muita huomioitavia yksityiskohtia maaperätekstuurien käytössä ovat puiden sijainnit ja niiden vaikutus ympäröivään maastoon. Kuolleet lehdet lisäävät maaperän monipuolisuutta ja näitä varten on luotu tekstuureja, joita maalataan kyseisten puiden alle ja lähietäisyyteen.

**Kiveä ja avokalliota** varten käytetään Unified Detail Mapping-menetelmää, jonka tarkoituksena on imitoida kivien ja kallioiden suurempia yksityiskohtia. Unified Detail Map -tiedosto voidaan käyttää yksittäisiin pienempiin kivi-objekteihin tai kallio-kokonaisuuksiin, joissa on useampi kivi yhdistettynä. Näiden lisäksi käytössä on diffuse-teksturi normal mapilla ja specular mapilla, joka sisältää gloss mapin. Nämä tekstuurit ovat sidoksissa toisiinsa ja luovat yksityiskohtaisen kokonaisuuden kiven pinnalla.

Specular map tehdään muuttamalla diffuusitekstuurista luodusta normal mapista mustavalkoinen teksturi. Specular mapin mustavalkoisuus määritellään valoisuuden mukaan. Alphakanavaan sijoitettu gloss map saadaan samasta normal mapista muuttamalla se mustavalkoiseksi tekstuurin kirkkauden mukaan (GIMP 2014).

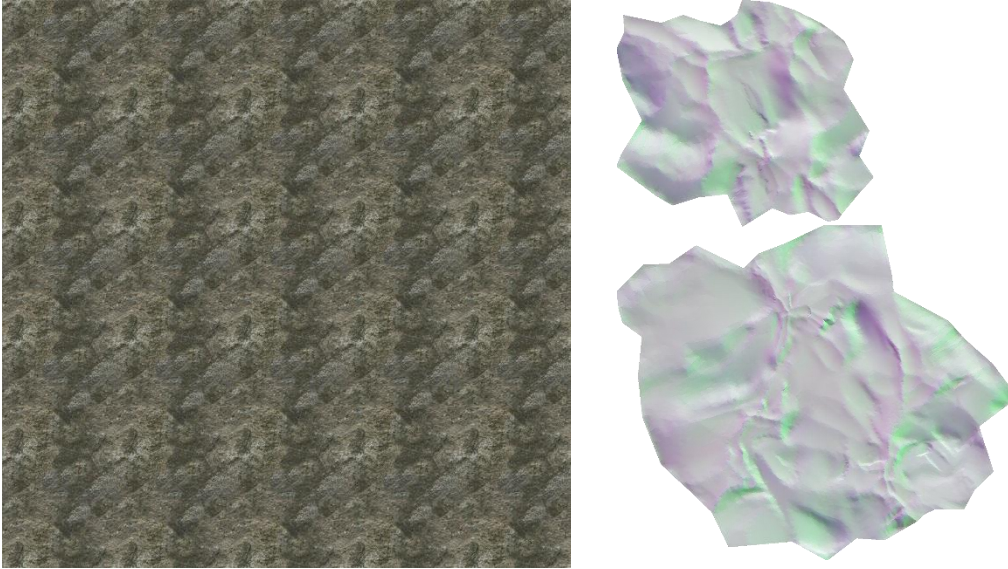
Unified Detail Mapping-menetelmässä mukaillaan Cryengine-dokumentaation mukaista järjestelyä, jota sovelletaan GIMP-kuvankäsittelysovellukselle. Menetelmää varten luodaan Ambient Occlusion (diffuse), Normal map ja Gloss map –tekstuurit.



Kuva 9. Esimerkki Unified Detail Mapissa käytettävistä kerroksista.

Aluksi Normal Map puretaan värikanavakohtaisiksi harmaasävykerroksiksi. GIMP:ssä luodaan tästä uusi tiedosto, jossa normal mapin punainen, vihreä ja sininen ovat omilla kerroksillaan harmaasävyisinä. Tiedostoon lisätään xNormal-sovelluksella luodut Ambient Occlusion ja Curvature mapit. Curvature map toimii Unified Detail Mapin Gloss mappina. Ambient occlusionin kirkkautta on vähennetty puoleen, jotta se toimisi paremmin Cryenginessä. Lisättyjen kerrosten jälkeen näistä laaditaan GIMP:ssä uusi RGBA –tiedosto (Kuva 9.), jossa punaisessa värikanavassa on Ambient Occlusion, vihreässä kanavassa on normal mapin punaiset harmaasävyarvot, sinisessä on gloss map ja alpha-kanavassa on normal mapin vihreät harmaasävyarvot (Cryengine 2014).





Kuva 10. Vasemmalla on hahmotelma tekstuurista viisinkertaisena tiilirakenteena objektia varten ja oikealla näkyy käytettävä UDM-tekstuuritiedosto.

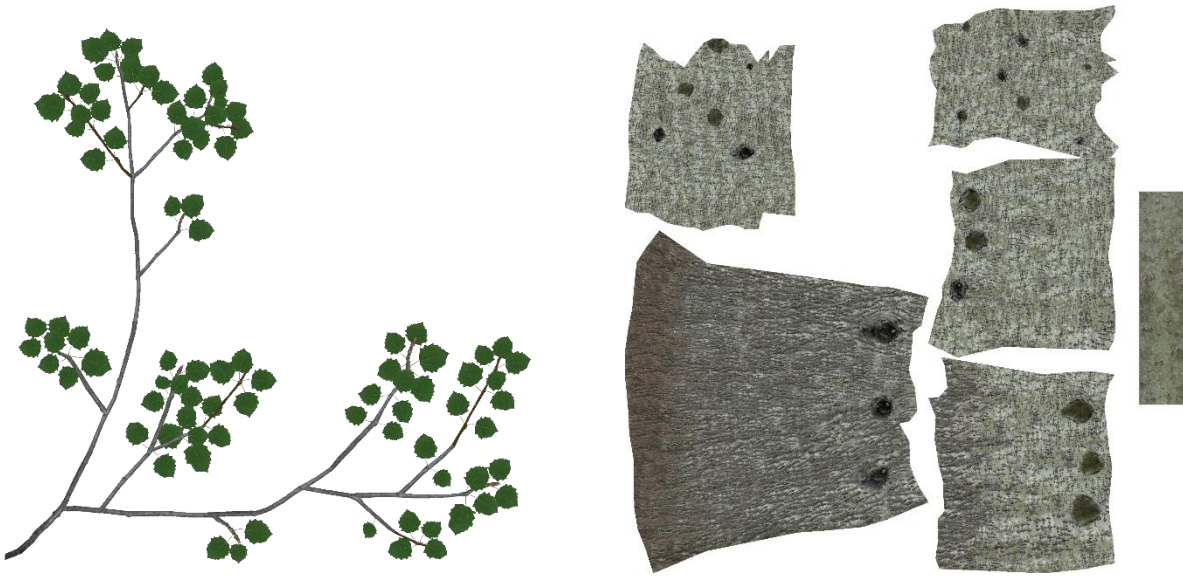
Cryengineen siirryttäessä mallinnettuun kiveen asetetaan Detail mappiin Unified Detail map-tiedosto, jonka koko on 1/x suhteessa diffuusi-tekstuurin toistettuun määrään (Kuva 10.). Tällä korjataan diffuse-tekstuurin toiston vaikutus verrattuna suuriin yksityiskohtiin käytettyyn normal mappiin. Diffuse-tekstuurin toistuvuudella kivistä on mahdollista luoda myös tarkemman näköisiä kuluttamatta merkittävästi näytönohjaimen muistia.

**Aluskasvillisuutta** varten on käytössä yksi 2k-alphatekstuuri. Tekstuuri sisältää useamman eri lajikkeen ja nämä ovat sijoitettu vapaasti käytettävän tekstuuritilan mukaisesti (Kuva 11.). Kaikki tässä tekstuurissa olevat lajikkeet ovat luonteeltaan yksinkertaisia kolmiulotteisia malleja, joissa käytetään mahdollisimman vähän polygoneja. Polygoneja käytetään mahdollisimman vähän siksi, koska aluskasvillisuudessa tähdätään kasvillisuuden runsauteen, jotta lopputulos näyttäisi luonnolliselta.



Kuva 11. Aluskasvillisuudessa käytetty alphatekstuuri.

Aluskasvillisuudessa keskitytään runsauden lisäksi monimuotoisuuteen, joka saavutetaan luomalla pelimateriaalia, joka sisältää vallitsevan lajikkeen lisäksi muita sille alueelle tyypillisiä lajeja. Nämä sijoitetaan jokaiseen pelimateriaalin vallitsevan lajin sekaan. Jokaisessa pelimateriaalissa on tärkeää saada niistä riittävän moniulotteisia, jotta ne ovat toistettavissa pelaajan sitä huomaamatta liian helposti. Tämä saavutetaan rikkomalla mahdolliset kuviot eri meshien välillä.



Kuva 12. Esimerkki puun tekstuureista.

**Puu** koostuu kahdesta tekstuurista (Kuva 12.). Runko koostetaan yhdestä isosta oksasta, joka on modulaarisesti paloittelavissa tarkempaa käyttöä varten. Puuta varten luodaan isompia oksakokonaisuuksia, jotka sijoitetaan puun rungon ympärille, joka on pelissä fyysisesti kiinteä osa puuta.

Rungon diffuusi-tekstuuri luodaan Blenderin texturepaintilla, jossa käytetään brusheina tekstuureja, joista puun runko koostuu. Tämä vähentää työtaakkaa merkittävästi runkoa teksturoitaessa. Näin lopputuloksesta tulee monipuolinen ja luonnollisemman näköinen.

Oksiin kytketään lehtiä ja pienempiä oksakokonaisuuksia, jotka koostuvat yhdestä alpha-tekstuurista. Tekstuuri on luonteeltaan modulaarinen ja näistä kaksiulotteisista oksista on mahdollista luoda isompia kokonaisuuksia, joista ei erota tekstuurin toistuvaa käyttöä.

Näillä tuotoksilla ja menetelmillä saatiin aikaan osa projektin tavoitteista. Projektissa tehdystä materiaalista on mahdollista luoda luonnonmukaisia kenttiä. Maaperä-tekstuureja on yhteensä kolmea erilaista. Jokaisessa näistä on käytössä normal map – ja heightmap –tekstuurit. Näihin kuuluvat neulaspinna, kuusipuun alunen ja sammalpinna. Kallioita ja kiviä varten veistettiin kolme erilaista kivikokonaisuutta. Jokainen näistä sisältää Unified Detail Mapping- tekniikan mukaisen tekstuuritiedoston. Näiden lisäksi

kiville luotiin kolme erilaista diffuusitekstuuria ja niihin lisättiin myös tekstuurikuvan mukaiset normal map –tekstuurit. Näihin lisättiin myös specular map –tekstuuuri gloss mapilla alphakanavassa. Aluskasvillisuutta varten luotiin yksi 2k-alphatekstuuuri, jonka pohjalta luotiin jokaista kasvilajia kohden yksi objekti. Alphatekstuurista voidaan päätellä, että tekstuuriin on vielä lisättävissä muitakin alueella esiintyvien kasvien tekstuureja. Puulajikkeita pelille luotiin kaksi pääalajia, joita alueella esiintyi. Nämä ovat kuusi ja haapa. Näitä varten käytössä on puun runkotekstuuuri ja oksien alphatekstuurit.

## 4 JOHTOPÄÄTÖS

Projektin alkuperäisenä tavoitteena oli luoda kokonaisvaltainen kenttä, jossa Cryengin ominaisuuksia käytetään kattavasti ja luonnon monimutkaisuutta tuodaan esille tehokkaasti pelien kenttäsuunnittelua silmälläpitäen. Tavoitteisiin ei ole päästy tämän opinnäytetyön yhteydessä. Myös avoimen lähdekoodiin perustuvien sovellusten rajoitteet tulivat selkeästi esille Cryenginellä työskennellessä. Työkaluja kuitenkin käytettiin kattavasti ja opinnäytetyön yhteydessä opittiin paljon henkilökohtaisia taitoja pelinkehittämiseen, kuvankäsittelyyn ja mallinnukseen liittyen.

Projektin heikoin osa-alue oli ajan käyttö ja mukautuminen haastaviin olosuhteisiin. Mallintaminen peleille oli aluksi verraten tuntematonta ja asioihin syventyminen vei usein enemmän aikaa, kuin alun perin arvioitiin. Monessa tilanteessa asioiden hallitseminen tarvittavalla tavalla vaati useamman kerran kokeiluja, joka teki opinnäytetyön alkuperäisen aihepiirin hahmottamisen vaikeaksi.

Yhtenä tavoitteena oli käyttää avoimeen lähdekoodiin perustuvia ilmaistyyökaluja. GIMP kuvankäsittelyohjelman valinta rajoitti tekstuurien käyttömahdollisuuksien suhteen. Suurin vaikutus kohdistui RGB -värikohtaiseen bittivaruuteen, joka on GIMP:llä 8 bittiä pikselin väriä kohden. Tämän johdosta Cryengine ei tarjonnut korkealaatuista tekstuuritiedoston muunnosta normal mapin parhaaseen laatuun, joka vaati 16 -bittiä väriä kohden. Tähän työkaluksi olisi sopinut lähinnä Photoshop kuvankäsittelyohjelma.

Alueen kasvillisuutta ei ole kyetty mallintamaan niin laaja-alaisesti, kuin alunperin suunniteltiin. Tekstuurikuvauksen ja mallintamisen haasteet veivät paljon aikaa varsinaisen materiaalin tuottamisesta. Tekstuurikuvien ottaminen vaati useamman kokeilun ennen sopivan kaltaisten kuvien löytymistä.

Näiden lisäksi puutteellisiksi ratkaisuihin osoittautuivat Meshlabin käyttö pääpainotteisena keinona luoda pienemmän polygonimäärän objekteja. Lopputuloksen polygonimäärät yksityiskohtien säilyttämiseen nähden ei onnistunut toivotun alhaisella polygonimäärällä.

Käsin uudelleenmuotoilu vaikutti edelleen paremmalta ratkaisulta. Tätä menetelmää noudatettiin puun runkoa luotaessa. Menetelmien yhteen sovittaminen saattaisi toimia sopivana aikaratkaisuna jatkossa. Näin tehtynä Meshlabissa luotua objektia ei tarvitse tehdä käsin alusta asti, vaan siihen voidaan tehdä tarvittavat korjaukset Blenderissä samalla kun polygonimäärää on mahdollista vähentää vielä vähän enemmän.

Mallintamisen osalta myös puiden lehtien asettaminen rungon ympärille vaatii vielä luonnollisen ratkaisun löytämistä. Nykyisen menetelmän ongelmaksi on osoittautunut lehtien runsauden puuttuminen. Puu näyttää isometrisestä kuvakulmasta toimivalta, mutta ensimmäisen persoonan näkökulmasta harvalehtiseltä.

Kenttäsuunnittelussa on huomioitu Cryenginelle varatut graafiset resurssit. Objekteja on yhdistetty graafisiksi kokonaisuuksiksi, jotta tämä vähentäisi tarvittavia piirtokutsuja ja vähentäisi muistin käyttöä. Kasvillisuudessa voi havaita vallitsevan puuston vaikutukset ympäröivien lajien kirjoon. Kenttäsuunnittelussa on myös pyritty huomioimaan pelien alueiden huomiopisteitä rajatuilla resursseilla.

Näiden periaatteiden käyttö on kuitenkin alkeellista. Kentässä ei ole käytössä yksityiskohtien asteittain skaalautuvuutta (Level of Detail) ollenkaan. Näiden vaikutusta suorituskykyyn on vaikeata arvioida nykyisen opinnäytekokonaisuuden pohjalta. Asteittain skaalautuvien yksityiskohtien käyttöönotto tarkoittaisi joka tapauksessa piirtokutsujen lisäämistä kokonaisuuteen.

Kuten edellä olevasta tekstistä voi päätellä. Kenttä on tässä vaiheessa keskeneräinen ja sen loppuunsaattaminen on ensisijainen tavoite tulevaisuudessa. Työkalut ja menetelmät ovat tämän projektin myötä tuttuja ja kenttäsuunnitteluun syventyminen jatkuu opinnäytetyön jälkeenkin. Tästä eteenpäin tarkoituksena on luoda lisää portfolioon soveltuvaa materiaalia sekä laajentaa ja soveltaa opinnäytetyössä saavutettua tietotaitoa.

Projektin sisältö on pintaraapaisu aihepiirin laajuuteen nähden. Laajempi syventyminen kenttäsuunnittelun eri osa-alueisiin tarinankerronnan ja pelimekaniikoiden kautta tuntuu luonnolliselta askeleelta eteenpäin. Kokonaisuuden hallinta vaatii kuitenkin määrätietoista työtä, jossa eri osa-alueet hahmotetaan oikealla tavalla ja ymmärretään

niiden tekemiseen kuluvaan aikaan. Opinnäytetyön keskittyessä kenttäsuunnittelun graafiseen materiaaliin ja on rajoittunut metsän mallintamiseen, joten uusien ja erilaisten alueiden mallintaminen on myös yksi kehittämisen kohteista.

Mallintamisen taitoja on mahdollista soveltaa myös muilla aloilla, joten tarkempi syventyminen kenttäsuunnittelun ulkopuolelle on täysin mahdollista. Rakennusten mallintamisen voi nähdä vielä osana kenttäsuunnittelua, mutta ihmishahmojen ja eläinten mallintaminen vaatii jo tarkempaa ymmärrystä anatomian lisäksi myös animaation tekemisestä.

Tekstuurien tekeminen valokuvien avulla vaatii omanlaisen lähestymistapansa. Tämä ei tarjonnut vielä tässä vaiheessa hirveästi taiteellisia vapauksia, mutta antoi tilaisuuden ymmärtää valokuvauksen ja kuvankäsittelyn keinoin, miten valo käyttäytyy eri pinnoilla ja minkälaisia haasteita se tarjosi tekstuurien tekemisessä. Jatkossa pintojen yksityiskohdat olisi tarkoitus pilkkoa pienempiin osa-alueisiin, jolloin niiden käyttöä voidaan miettiä paremmin tilanteiden tarpeen mukaan.

Kenttäsuunnittelu on kokonaisvaltainen osa-alue pelinkehitystä, jossa pitää ymmärtää laaja-alaisesti pelin tavoitteita. Tämä korostuu etenkin indie -pelinkehityksessä. Aiemmin saatavilla oli vain kalliita mallinnus- ja kuvankäsittelylisenssejä graafisen materiaalin luontia varten. Tähän nähden työkalujen saatavuus on parantunut merkittävästi ja nykyään graafisesti vaativienkin pelien kehittäminen on mahdollista pienelläkin budjetilla.

## 5 LÄHTEET

Blender 2014. Blender Foundation. Viitattu 12.8.2014 <http://www.blender.org/>

Blender Wiki 2015. Blender 2.66: Dynamic Topology Sculpting . Viitattu 2.4.2015  
[http://wiki.blender.org/index.php/Dev:Ref/Release\\_Notes/2.66/Dynamic\\_Topology\\_Sculpting](http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.66/Dynamic_Topology_Sculpting)

CG Textures, 2007. Recovering detailed normals from photo textures. By Scott Warren.  
Viitattu 7.12.2014  
<http://www.cgtextures.com/content.php?action=tutorial&name=normalmap>

Cryengine 2013. Unified Detail Mapping. Viitattu 8.12.2014  
<http://docs.cryengine.com/display/SDKDOC2/Unified+Detail+Mapping>

Cryengine 2014. 2013 Crytek Oy. Viitattu 12.8.2014 <http://cryengine.com/features>

Cryengine 2014. Art Pipeline. Viitattu 10.4.2015  
<http://docs.cryengine.com/display/SDKDOC2/Art+Pipeline>

Cryengine 2014. Drawcalls. Viitattu 8.12.2014  
<http://docs.cryengine.com/display/SDKDOC2/Asset+Performance+Guidelines>

DxOMark 2014. Tests and reviews for the camera Canon EOS 60D. Viitattu 8.12.2014  
<http://www.dxomark.com/Cameras/Canon/EOS-60D>

GIMP 2014. Desaturate. Viitattu 5.3.2015 <http://docs.gimp.org/en/gimp-tool-desaturate.html>

Jenssen, M. 2011. Single Player Level Design Workflow. Viitattu 4.3.2015  
[http://worldofleveldesign.com/categories/wold-members-tutorials/magnar\\_jenssen/single-player-level-design-workflow-magnar-jenssen.php](http://worldofleveldesign.com/categories/wold-members-tutorials/magnar_jenssen/single-player-level-design-workflow-magnar-jenssen.php)

Palamar, T. 2012. Mastering Autodesk Maya 2013. Indianapolis, Indiana: John Wiley & Sons, Inc.



Photozone 2007. Canon EF-S 18-55mm f/3.5-5.6 IS - Review / Test Report – Analysis. Viitattu 4.3.2015 <http://www.photozone.de/canon-eos/181-canon-ef-s-18-55mm-f35-56-is-test-report--review?start=1>

Shapeways 2015. Polygon Reduction with Meshlab. Viitattu 8.4.2015 [http://www.shapeways.com/tutorials/polygon\\_reduction\\_with\\_meshlab](http://www.shapeways.com/tutorials/polygon_reduction_with_meshlab)

Toledo, P. 2010. Brief Consideration About Materials. Viitattu 9.6.2014 <http://www.manufato.com/?p=902>

Tom's Hardware 2015. Best Gaming CPUs For The Money: March 2015. Viitattu 4.3.2015 <http://www.tomshardware.com/reviews/gaming-cpu-review-overclock,3106-5.html>

Tom's Hardware 2015. Best Graphics Cards For The Money: March 2015. Viitattu 4.3.2015 <http://www.tomshardware.com/reviews/gaming-graphics-card-review,3107-7.html>

Ward, A. 2008. Game Character Development: Digital Sculpting for the Realtime Artist. Boston, MA, USA: Course Technology.

Wikipedia 2015. Digital camera ISO speed and exposure index. Viitattu 16.3.2015 [https://en.wikipedia.org/wiki/Film\\_speed#Digital\\_camera\\_ISO\\_speed\\_and\\_exposure\\_index](https://en.wikipedia.org/wiki/Film_speed#Digital_camera_ISO_speed_and_exposure_index)

Wikipedia 2015. Aperture. Viitattu 16.3.2015 <https://en.wikipedia.org/wiki/Aperture>